Trust-Aware RBAC

Vladimir Oleshchuk

Department of ICT, University of Agder
PB 509, N-4898 Grimstad, Norway
vladimir.oleshchuk@uia.no

Abstract. In this paper we propose a trust-aware enhancement of RBAC (TA-RBAC) that takes trustworthiness of users into consideration explicitly before granting access. We assume that each role in the framework is associated with an expression that describe trustworthiness of subjects required to be able to activate the role, and each subject (user) has assigned trustworthiness level in the system. By adding trustworthiness constraints to roles we enhance system, for example, with more flexible ability to delegate roles, to control reading/updating of objects by denying such operations to those subjects that violate trustworthiness requirements.

1 Introduction

Over the years, Role-Based Access Control (RBAC) has established itself as a generalized approach for handling access control in computer systems and differs from traditional identity based access control models in that it takes advantage of the concept of role relations [11, 10]. For these models, access to data and resources are based on the organizational activities and responsibilities, or roles, which users possess in a system. In RBAC, a user's ability to access computer resources (objects) is determined by the user's association with roles and by these roles' permissions to perform operations on objects. Usually roles correspond to different job functions within an organization. Job functions are associated sets of permissions, which can be seen as expression of trustworthiness of role holder within an organization.

Different access control models to support various organizational security policies have been proposed over the years. The first model that supported integrity protection of resources was Biba Model developed by Kenneth J. Biba in 1977 [5]. This model describes a set of access control rules designed to ensure data integrity. The idea is that subjects on lower integrity levels are not permitted to modify (corrupt) objects on higher integrity levels (known as "no write up" rule). Correspondently, subjects on higher integrity levels can be corrupted by accessing objects on lower integrity levels (known as "no read down" rule). This model can be considered as one of the first models dealing with trustworthiness (also implicitly). According to [6], "integrity refers to the trustworthiness of data or resources, and it is usually phrased in terms of preventing improper or unauthorized change".

Bell-LaPadula model was proposed to support protection of confidentiality [3]. This model describes access rules to protect confidentiality of resources.

I. Kotenko and V. Skormin (Eds.): MMM-ACNS 2012, LNCS 7531, pp. 97–107, 2012.

Assuming that subjects are assigned clearance levels and objects are assigned classification levels the model's rule support no write down and no read up rules. However these rules also prevent effective communication.

Role Based Access Control (RBAC) Model has been found to be quite useful and has drawn a lot of research interest over the last fifteen years. It was recently defined as NIST/ANSI Standard [2]. Traditional RBAC considers user to role as well as role to permission assignments to be static in nature with respect to space and time. However it was observed that the context-aware access should play a more active role in access decision process. For example, in mobile applications, spatial context plays an increasingly important role both in defining and enforcing more elaborated security policies since in many applications locations of participants should directly influence access control decisions [1, 7, 9, 17–19, 22].

In recent years some new extensions of RBAC that use notion of trust have being proposed [4, 8, 24, 25]. Trust-aware access control models are more suitable for decentralized, multi-centric systems with dynamic population of users where traditional access models do not work well. One of the benefit of trust awareness (not considered yet in existing extensions of RBAC) is an ability to provide history-based solutions.

In this paper we propose to enhance the discrete trust paradigm with more elaborated multi-level trust paradigm based on notion of opinion from subjective logic. In this new trust-enhanced model, trust levels of subjects (human users or software agents acting on users behalf), roles and objects (data, programs, processes) are expressed as opinions about their trustworthiness determined by history of interactions between subjects (users) and objects. We define such opinions in the framework of subjective logic [14, 15].

The rest of the paper is organized as follows. In Section 2 we provide brief review of related work. We introduce notation and notions of subjective logic in Section 3. Then we present our trust-enhanced RBAC model in Section 4. Finally, Section 5 concludes the paper.

2 Related Work

Recent years many context-aware extensions of RBAC were proposed. Most of them considered specifically access wth respect of time and location.

In [17, 18], authors extend the RBAC model by specifying spatial restrictions on permissions assigned to roles which enables a role to have permissions assigned to it dependent on the location. Spatial constraints on permissions assigned to a role can be beneficial when specifying the access control policy in mobile environments where the location from which a user wnats to access services is a key security parameter [19]. Authors have extended the RBAC model, and introduced a formal model that allows specifying spatial constraints on permissions associated with roles in different locations. In [9] authors propose a spatially-aware RBAC model called GEO-RBAC. In the proposed approach authors propose the notion of spatial roles which are defined as roles with spatial extents defining the boundaries of the space in which the role can be used by the

users. In this approach roles are activated based on the position of the user. Another location aware RBAC model has been proposed in [22]. Authors show how the different components of the core RBAC model are related to location, how existing operations need to be modified and what new operations are needed. They left elaboration of role hierarchies and separation of duty constraints for future work. Several authors have also proposed models that combine both spatial and temporal aspects [7, 1].

However, relatively recently some authors started to consider trust as a new context parameter. This is motivated by desire to expand RBAC models to meet security challenges posed by new application paradigms were existing models found to be inadequate (for example, for open and decentralized systems or mobile and pervasive systems). In [4] authors motivation was to provide access control model for Web services. They propose an extended, trust-enhanced version of XML-based RBAC (X-RBAC) framework that incorporates context-based access control. In the framework, the authors rely on certification provided by trusted third party assigning levels of trust to users. In [8] authors propose a trust based access control model called TrustBAC by extending the conventional RBAC model with the notion of trust levels. Users are assigned to trust levels instead of roles based on a number of factors like user credentials, user behavior history, user recommendation etc. Trust levels are numbers between -1 and $+1$ and which computes by the trust evaluation module. In [24] authors propose a trust-based RBAC model for pervasive computing systems where they adapt trust model they proposed earlier [23] to evaluate trustworthiness. Needs for delegation arise in many applications. Trustworthy delegation that does not violate security policies by allowing access only to trustworthy delegatee was considered in [25]. In [20] authors propose a framework that combines strenths of RBAC systhems and trust-management systems to deal with access control in decenralized collabarative systems. The trustworthness of subjects is determined on the base of their certified attributes.

3 Measurement of Trust: Subjective Logic

In this section, we show how to express the levels of trustworthiness in the framework of subjective logic. Following [14, 15] we first define the term opinion, denoted ω, that expresses opinion about level of trustworthiness.

Let t, d and u be such that $t + d + u = 1$ and $t, d, u \in [0, 1]$. Then a triple $\omega = \{t, d, u\}$ is called an opinion, where components t, d and u represent levels of trust, distrust and uncertainty respectively. The levels of trustworthiness are expressed by opinions. Varying these parameters, we can express different levels of trustworthiness. Expressing trustworthiness using three values instead of just one trust level provides a more adequate trust model of real world with uncertainties. These parameters are not treated equally when different opinions are combined.

The subjective logic defines a set of logical operators for combining opinions including conjunction, recommendation, and consensus. For more details related to subjective logic the reader is recommended to consult [14–16].

Let $\omega^A = \{t^A, d^A, u^A\}$ denote an opinion about trustworthiness of entity A.

Let $\omega_p^A = \{t_p^A, d_p^A, u_p^A\}$ denote an opinion of entity A about consequences for security of an action p. In context of this paper, A can be an RBAC system itself or a user, and an action p may be "activate role r". Assume that an entity A has an opinion $\omega_p^A = \{t_p^A, d_p^A, u_p^A\}$ about potential security threat of p, and an opinion $\omega_q^A = \{t_q^A, d_q^A, u_q^A\}$ about potential security threat q. Then A's opinion about consequences for security of both actions, denoted as $p \wedge q$, can be found (according to [14]) as following:

$$\omega_{p\wedge q}^A = \omega_p^A \wedge \omega_q^A = \{t_{p\wedge q}^A, d_{p\wedge q}^A, u_{p\wedge q}^A\}$$

where

$$t_{p\wedge q}^A = t_p^A t_q^A$$
$$d_{p\wedge q}^A = d_p^A + d_q^A - d_p^A d_q^A$$
$$u_{p\wedge q}^A = t_p^A u_q^A + u_p^A t_q^A + u_p^A u_q^A$$

Let A and B be two entities (RBAC systems or users). If A is a RBAC system itself, it has opinions about trustworthiness of its own users but not about users in other RBAC systems (in a federated system). Then $\omega_B^A = \{t_B^A, d_B^A, u_B^A\}$ denotes an opinion of A entity about trustworthiness of recommendations given by B. Assume B gives its recommendation to A about trustworthiness of action p in the form of its opinion ω_p^B. Assuming that an entity A does not have any direct opinion ω_p^A about p it will try to deduce some indirect opinion about trustworthiness of p, denoted ω_p^{AB}, based on recommendation given by B. For this purpose the recommendation operator $\otimes$ is used (according to [14]) as follows:

$$\omega_p^{AB} = \omega_B^A \otimes \omega_p^B = \{t_p^{AB}, d_p^{AB}, u_p^{AB}\}$$

where

$$t_p^{AB} = t_B^A t_p^B$$
$$d_p^{AB} = t_B^A d_p^B$$
$$u_p^{AB} = d_B^A + u_B^A + t_B^A u_p^B$$

In context of this work there is a need to combine independent opinions about trustworthiness of the same action. According to [16], "The consensus opinion of two possibly conflicting argument opinions is an opinion that reflects both argument opinions in a fair and equal way". Adjusting reasoning from [16] we can argue applicability of the consensus operator (defined below).

Let A and B be two entities that represent entities such as the system or users. Let $\omega^A = \{t^A, d^A, u^A\}$ and $\omega^B = \{t^B, d^B, u^B\}$ be two opinions of A and B about the same action (object, user). In the case when there are several independent opinions about the same action, subjective logic suggests to use a consensus operator $\oplus$ to combine these independent opinions. According to subjective logic, the combined consensus opinion ω based on ω^A and ω^B is defined as follows:

$$\omega = \omega^A \oplus \omega^B$$

where

$$t = (t^A u^B + t^B u^A)/(u^A + u^B - u^A u^B)$$
$$d = \left(d^A u^B + d^B u^A\right) / \left(u^A + u^B - u^A u^B\right) r$$
$$u = \left(u^A u^B\right) / \left(u^A + u^B - u^A u^B\right)$$

We define odering relation on opinions in the following way. We assume that opinion ω^A is more trustworthy than opinion ω^B, denoted $\omega^A \gg \omega^B$, if $t^A > t^B$. If $t^A = t^B$, then higher distrust value means lower corresponding uncertainty value. Assuming that decreasing uncertainty may contribute equally to both trust and distrust values, we choose opinions with higher uncertainty (and with equal trust values t^A and t^B) be more trustworthy. Formally, if $t^A = t^B$ then $\omega^A \gg \omega^B$ if $u^A > u^B$.

In the following section we describe how trustworthiness expressed as opinions can be integrated into traditional RBAC model.

4 Trust-Aware RBAC (TA-RBAC)

Informally, we assume that there is a set of roles $ROLES$ that may be assigned to users from $USERS$. Each user u may have many roles assigned at the same time. Each role r from $ROLES$ is associated with a pair $\{\omega^l, \omega^h\}$. It means that trustworthiness of a user u who are able to activate r cannot be lower than ω^l and higher than ω^h respectively.

In this section we propose a trust-aware RBAC (TA-RBAC) model that is an extension of traditional RBAC model [2]. Since the time of introduction of RBAC, various context-aware models were proposed (see Section 2). However some important features dictated by current and future real-world applications such as, for example, handling access request in proximity of specific devices [21] or taking history of behavior or actions are still not well-developed. This is the motivation behind the proposed extension.

Informally, in the proposed extension of RBAC each role r from $ROLES$ has assigned requirements on trustworthiness of users u from $USERS$ that are permitted to activate this role. It means that to have role assigned to u is not enough - it is also necessary verify that the current level of trustworthiness of u satisfies trustworthiness requirements assigned to the role r. This is an additional constraint (requirement) that may be used to take into account behavior history of u such as what roles u has activated in the past, for which purposes, from which location, when, etc. Dynamically changing trustworthiness constraints of roles provides additional constraints on ability of u to activate assigned roles, for example, in case of trust-aware separation of duties (explained in Subsection 4.3).

The proposed TA-RBAC model consists of the following five basic components: $USERS$, $ROLES$, $PRMS$, $SESSIONS$ and TRW representing the set of users, roles, permissions, sessions and opinions respectively, where TRW is a set of possible opinions about trustworthiness of users and roles. Users from $USERS$ are considered to be either humans, devices or processes operating on

behalf of other users that can access resources (services) to perform some actions. $ROLES$ describes a collection of roles defined as a set of permissions that may be guarded by trustworthiness constraints to control accessibility to resources (objects). $PRMS$ is a set of permissions to access resources/services to perform a specific action if trustworthiness of the user satisfies trust requirements. Elements of TRW is specified by means of subjective logic opinions.

4.1 Core Model

The model defines several functions and relations on the sets $USERS$, $ROLES$, $PRMS$, $SESSIONS$, TRI needed for specification and implementation of TA-RBAC. The user assignment relation UA represents the assignment of a user from $USERS$ to roles from $ROLES$. The permission assignment relation PA represents the assignment of permissions to roles based on trustworthiness of both users and services. Definition below gives formal descriptions of some important functions and relations.

Definition 1. *TA-RBAC core model consists of the following components:*

- $USERS$, $ROLES$, $PRMS$, $SESSIONS$ *and* TRW, *represent the finite sets of users, roles, permissions, sessions, and opinions respectively;*
- $PRMS = REQUESTS \times SERVICES$ *where* $REQUESTS$ *denotes all action requests users can send to services denoted as* $SERVICES$*;*
- TRW *represents trustworthiness in form of subjective logic opinions (including complete trust* $(1,0,0)$*, complete distrust* $(0,1,0)$ *and complete uncertainty* $(0,0,1)$*);*
- $UA \subseteq USERS \times ROLES$ *, the relation that associates users with roles;*
- $TRI \subseteq TRW \times TRW$ *represents the set of trustworthiness intervals, where* $(\omega_1, \omega_2) \in TRI$ *means* $\omega_1 \gg \omega_2$ *or* $\omega_1 = \omega_2$*;*
- $UT \subseteq USERS \times TRW$ *defines assignment of trustworthiness level to users;*
- $assigned\_trust\,(u : USERS) \rightarrow TRW$ *, the function mapping a user* u *into an opinion. Formally, trustworthiness of user* u *can be found as* $assigned\_trust\,(u) = \{\omega | (u, \omega) \in UT\}$*;*
- $RT \subseteq ROLES \times TRI$ *defines assignment of trustworthiness level to roles;*
- $role\_trust\_constr(r : ROLES) \rightarrow TRI$*, the function that maps a role* r *to trustworthiness interval from* TRI*. Formally, trustworthiness constraints associated with* r *can be found as* $role\_trust\_constr(r) = \{t | (r, t) \in RT\}$*.*
- $assigned\_users(r : ROLES) \rightarrow 2^{USERS}$ *, the mapping of a role onto a set of users. Formally, users assigned to role* r *can be found as* $assigned\_users(r) = \{u \in USERS | (u, r) \in UA\}$ *;*
- $assigned\_roles(u : USERS) \rightarrow 2^{ROLES}$ *, the mapping of a user* u *onto a set of roles. Formally, roles assigned to a user* u *can be found as* $assigned\_roles(u) = \{r \in ROLES | (u, r) \in UA\}$*;*
- $PA \subseteq ROLES \times PRMS$ *, the relation that defines what permissions* $PRMS$ *of a role* r *from* $ROLES$ *are available to a user with trustworthiness level suitable to activate* r*. That is* $(r, p) \in PA$ *means that if user* u *has assigned role* r *she can utilize permission* $p = (req, srv)$ *to access service* srv *when trustworthiness of* u *satisfies trust requirement of* r*.*

- *assigned_perms*$(r : ROLES, t : TRW) \rightarrow 2^{PRMS}$ *describes permissions assigned to role* r *when trustworthiness of a user* u *activating* r *satisfies trust requirements of* r*. Formally,* $assigned\_perms(r) = \{p|(r,p) \in PA\}$*;*
- *user_sessions*$(u : USERS) \rightarrow 2^{SESSIONS}$*, associate a user* u *with set of sessions;*
- *session_roles*$(s : SESSIONS) \rightarrow 2^{ROLES}$*, the mapping of session* s *to a set of roles;*
- *avail_session_perms*$(s : SESSIONS, t : TRI) \rightarrow 2^{PRMS}$ *, the permissions available in a session* s *when trust requirements satisfies* t*.*
  Formally, $avail\_session\_perms(s,t) = \bigcup_{r \in session\_roles(s)} assigned\_perms(r,t)$
- *auth_user*$(r : ROLES, t : TRI) \rightarrow USERS$ *identifies users assigned to role* r *satisfying* t*.*
- *auth_user*$(r_1, r_2, ... : ROLES) \rightarrow USERS$ *identifies users assigned to at least roles* $r_1, r_2, ..., r_k$*;*
- *auth_user*$(s : SESSIONS; r_1, r_2, ... : ROLES) \rightarrow USERS$ *identifies users that are authorized to activate simultaneously roles* $r_1, r_2, ...r_k$ *within a session* s*.*

We assume that each user u assigned an initial trust level, $init\_trust : USERS \rightarrow TRW$, and each role r from $ROLES$ has assigned trustworthiness constraints as $\left[\omega^l, \omega^h\right]$ from TRI where ω^l denotes the lowest trust level that a user u must have to be able activate r; ω^h denotes a trustworthiness level that user activating r must not exceed (in cases when it is not essential $\omega^h = (1,0,0)$, meaning highest possible trustworthiness).

4.2 Role Hierarchies

In order to extend the core TA-RBAC to Hierarchical TA-RBAC we need to define hierarchies and inheritance for roles in presence of trust describing how roles inherit permissions from their junior roles. Definition below formally introduces our solution.

Definition 2. *Relation* RH*, defines as* $RH \subseteq ROLES \times ROLES$*, is a partial order on roles, with respect to trustworthiness, called dominance relation, denoted as* $\succeq$ *, where* $r_i \succeq r_j$ *for* $r_i, r_j \in ROLES$ *means that* r_i *inherits permissions of* r_j*. Let* $role\_trust\_constr(r_j) = (\omega_j^l, \omega_j^h)$ *and* $role\_trust\_constr(r_i) = (\omega_i^l, \omega_i^h)$ *. We saying that role* r_i *inherits permissions of role* r_j *with trustworthiness constraints for user assigned to* r_i *computed as following:* $(\omega_j^l \oplus \omega_i^l, \omega_j^r \oplus \omega_i^r)$*.*

The use of the consensus operator can be argued that trustworthiness constraints (ω_i^l, ω_i^h) and (ω_j^l, ω_j^h) can be seen as independent opinions on trustworthiness constraints of r_j.

The reason for use of the consensus operator is as following. There are two independent sets of trustworthiness constraints of r_i: 1) the trusworthiness constraints on r_i independant of hierarchies, and 2) trustworthiness constraints derived from constraints of the inherited role r_j. In case of two independent

opinions the consensus operator usually applied to find a combined opnions that constitutes the new constraints. The user assigned to r_i have to satisfy native constraints of r_i to use its permissions. However when the user wants to use inherited permissions of r_j, she must satisfy constraints that are consensus between constraints of r_i and r_j. A user assigned directly to r_j have to satisfy native constraints of r_j in order to activate its permissions.

4.3 Separation of Duties

The efficiency of RBAC to enforce the principle of least privilege is partly due to ability to enforce Separation of Duties (SoD) principle. However by constraining ability of users to activate some combination of roles reduce system's usability, for example, in small organizations where number of users are small with respect to number of roles.

The trust-awareness may provide better flexibility by for example putting less constraints on highly trustful users. However, that means that the notion of SoD needs to be re-defined. We defines both Trust-aware Static SoD (TSSoD) and Trust-aware Dynamic SoD (TDSoD), where requirement to trustworthiness of a user who activates (partly) mutually exclusive roles increases. That is, two roles with assigned permissions may be partly mutually exclusive if system requires higher level of trustworthiness comparing to requirements when only each role will be activated separately.

We define trust-aware separation of duty SoD property as following set. Let $S = \{(r, \omega), ...\}$ where $r \in ROLES$, $\omega \in TRW$ and ω informally represents opinion of the RBAC system on how security sensitive activation of r is. When a user u activates a set of roles $r_1, r_2, ..., r_k$ such that from $\{(r_i, \omega_i) | i = 1, ..., k\} \subseteq S$ the requirements to trustworthiness of the user will be computed as $\omega_1 \wedge ... \wedge \omega_k$. Informally, it means that more trustworthy users are able to activate simultaneously more roles from S.

Formally, TSSoD can be defined as following.

Definition 3. *$TSSoD \in 2^{ROLES \times TRW}$ is a set of pairs (r, ω) where is a role, ω is a trustworthiness, with the property that no user can be assigned to subset of roles from TSSoD such that conjuction of opinions ω_i of assigned roles will exceed initial trustworthiness of that user. Formally, $\{(r_i, \omega_i) | i = 1, 2, ..., k\} \subseteq TSSoD \wedge \forall u \in auth\_user(r_i \, | i = 1, ..., k) \Rightarrow user\_trw(u) \gg \wedge_{i=1,2,...,k} \omega_i$*

Formally, TDSoD can be defined as following.

Definition 4. *$TDSoD \in 2^{ROLES \times TRW}$ is a collection of pairs (r, ω) where r is a role, ω is a trustworthiness, with the property that no user can activate a subset of roles from TDSoD such that conjunction of opinions ω_i of roles activated within a session will exceed initial trustworthiness of that activating user. Formally: $\{(r_i, \omega_i) \, | i = 1, 2, ..., k\} \subseteq TDSoD \wedge s \in user\_sessions(u) \Rightarrow \forall u \in auth\_user(s, r_i \, | i = 1, ..., k) \Rightarrow user\_trw(u) \gg \wedge_{i=1,2,...,k} \omega_i$*

The reason of using conjunction operator is as following. Since opinion about security implications (potential security threats) of activation role r_i is ω_i the

opinion about security implications of activation simultaniosly a set of such roles can be computed as a conjunction of coresponding opinions.

4.4 Delegation

Many authors have studied role delegation in RBAC [27, 26, 25]. In this work we propose to control the ability to delegate roles by taking into consideration trustworthiness of delegatee and trustworthiness constraints of delegated role. The idea is that a user with high degree of trustworthiness can delegate role with relatively high trustworthiness requirements to less trustful user (since combination of role constraints and user trustworthiness can decrease required trustworthiness and therefore make it available for less trustful user). Delegation is not a part of standart RBAC and it may result in security violation. However it may provide better usability.

Suppose that a user u wants to delegate her role r to another user u' (that has not this role assigned). One way to do this is to add the instance of r called r' (r' is a new instance of r which may differs from r by trustworthiness constraints) to u' by adding r' to the list of assigned roles assigned to u'. That is

$$assigned\_role(u') = assigned\_role(u') \cup \{r', (\omega^l_{r'}, \omega^h_{r'})\},$$

where the trustworthiness requirements $(\omega^l_{r'}, \omega^h_{r'})$ on this delegated role r' will be computed as combination of trustworthiness of u and constraints of r as following (since it can be seen as a recommendation of r by u to u') as following:

$$\omega^l_{r'} = \omega_u \otimes \omega^l_r$$

$$\omega^h_{r'} = \omega_u \otimes \omega^h_r$$

where ω_u denotes trustworthiness of u. The use the recommendation operator in this case because we consider delegation of role r by u to u' as a recommendation of u to the system to assign r to u'. The system uses trusworthiness of u as trustworthiness of u's recommendations and computes trustworthiness constraints for r' by taking into consideration of trustworthiness of u.

5 Conclusion

In this work we propose a novel trust-aware RBAC model (TA-RBAC). Our approach integrates trustworthiness levels expressed as opinions in subjective logic with traditional RBAC model. We have defined trust-aware role inheritance which is essential for defining Hierarchical trust-aware RBAC. By using subjective logic operations for combining independent opinions we define static and dynamic trust-aware SoD. We use recommendation operator to define trust-aware role delegation.

References

1. Aich, S., Sural, S., Majumdar, A.: STARBAC: Spatiotemporal Role Based Access Control. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part II. LNCS, vol. 4804, pp. 1567–1582. Springer, Heidelberg (2007)
2. ANSI/INCITS 359-2004. Role Based Access Control. InterNational Committee for Information Technology Standards (formerly NCITS) / 03-Feb-2004 / 56 pages
3. Bell, D.E., LaPadula, L.J.: Secure Computer Systems: Mathematical Foundations. MITRE Corporation (1973)
4. Bhatti, R., Bertino, E., Ghafoor, A.: A Trust-Based Context-Aware Access Control Model for Web-Services, Distributed and Parallel Databases (2005)
5. Biba, K.J.: Integrity Considerations for Secure Computer Systems, MTR-3153, The Mitre Corporation (April 1977)
6. Bishop, M.: Computer Security: Art and Science. Addison Wesley, Boston (2003)
7. Chandran, S.M., Joshi, J.B.D.: LoT-RBAC: A Location and Time-Based RBAC Model. In: Ngu, A.H.H., Kitsuregawa, M., Neuhold, E.J., Chung, J.-Y., Sheng, Q.Z. (eds.) WISE 2005. LNCS, vol. 3806, pp. 361–375. Springer, Heidelberg (2005)
8. Chakraborty, S., Ray, I.: TrustBAC: integrating trust relationships into the RBAC model for access control in open systems. In: Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies (SACMAT 2006), pp. 49–58. ACM, New York (2006)
9. Damiani, M.L., Bertino, E., Catania, B., Perlasca, P.: Geo-RBAC: A spatially aware RBAC. ACM Trans. Inf. Syst. Secur. 10, 1–42
10. Ferraiolo, D.F., Kuhn, D.R., Chandramouli, R.: Role-Based Access Control. Artech House (2003)
11. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC) 4(3), 224–274 (2001)
12. Ferreira, A., Chadwick, D., Farinha, P., Correia, R., Zao, G., Chilro, R., Antunes, L.: How to securely break into RBAC: The BTG-RBAC model. In: Annual Computer Security Applications Conference, ACSAC 2009, pp. 23–31 (December 2009)
13. Ferreira, A., Cruz-Correia, R., Antunes, L., Farinha, P., Oliveira-Palhares, E., Chadwick, D., Costa-Pereira, A.: How to break access control in a controlled manner. In: 19th IEEE International Symposium on Computer-Based Medical Systems CBMS 2006, pp. 847–854 (2006)
14. Jøsang, A.: An Algebra for Assessing Trust in Certification Chains. In: Kochmar, J. (ed.) Proceedings of the Networks and Distributed Systems Security, NDSS 1999 (1999)
15. Jøsang, A.: A Logic of Uncertain Probabilities, International Journal of Uncertainty. Fuzziness and Knowledge-Based Systems 9(3), 279–311 (2001)
16. Jøsang, A.: The Consensus Operator for Combining Beliefs. Artificial Intelligence Journal 142(1-2), 157–170 (2002)
17. Hansen, F., Oleshchuk, V.: Spatial role-based access control model for wireless networks. In: IEEE Vehicular Technology Conference VTC 2003, vol. 3, pp. 2093–2097 (2003)
18. Hansen, F., Oleshchuk, V.: SRBAC: A spatial role-based access control model for mobile systems. In: Proceedings of the Seventh Nordic Workshop on Secure IT Systems (Nordsec 2003), October 15-17, pp. 129–141 (2003)

19. Hansen, F., Oleshchuk, V.: Location-based security framework for use of handheld devices in medical information systems. In: Fourth Annual IEEE International Conference on Pervasive Computing and Communications, PerCom Workshops 2006, March 13-17, pp. 564–569 (2006)
20. Li, N., Mitchell, J.C., Winsborough, W.H.: Design of a role-based trust management framework. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy, pp. 114–130. IEEE Computer Society Press (2002)
21. Oleshchuk, V., Fensli, R.: Remote patient monitoring within a future 5G infrastructure. Wireless Personal Communications 57, 431–439
22. Ray, I., Kumar, M., Yu, L.: LRBAC: A Location-Aware Role-Based Access Control Model. In: Bagchi, A., Atluri, V. (eds.) ICISS 2006. LNCS, vol. 4332, pp. 147–161. Springer, Heidelberg (2006)
23. Ray, I., Ray, I., Chakraborty, S.: An interoperable context sensitive model of trust. Journal of Intelligent Information Systems 32(1), 75–104 (2009)
24. Toahchoodee, M., Abdunabi, R., Ray, I., Ray, I.: A Trust-Based Access Control Model for Pervasive Computing Applications. In: Gudes, E., Vaidya, J. (eds.) Data and Applications Security XXIII. LNCS, vol. 5645, pp. 307–314. Springer, Heidelberg (2009)
25. Toahchoodee, M., Xie, X., Ray, I.: Towards Trustworthy Delegation in Role-Based Access Control Model. In: Proceedings of the 12th International Conference on Information Security, Pisa, Italy, September 07-09 (2009)
26. Wainer, J., Kumar, A.: A fine-grained, controllable, user-to-user delegation method in RBAC. In: Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies (SACMAT 2005), pp. 59–66. ACM, New York (2005)
27. Zhang, X., Oh, S., Sandhu, R.: PBDM: a flexible delegation model in RBAC. In: Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies (SACMAT 2003), pp. 149–157. ACM, New York (2003)